



digital
vision
group

TRIPnxp v8.4-1

Change History

2026-05-06

Copyright (c) 2026 DVG Operations GmbH

Published By:
DVG Operations GmbH, Hilpertstraße 20, 64295 Darmstadt, Germany
E-mail info@dvg.services, www.digitalvision-group.com

End User License Agreement

All rights to this software, its documentation and logotypes of the TRIP product family and software (altogether "Software") supplied by DVG Operations GmbH (DVG) are exclusively owned by DVG.

The transfer of this Software, solutions or parts thereof requires the prior written agreement of DVG. Furthermore, the customer has the right to use licensed Software and / or process solutions supplied by DVG to the extent specified in his contract with DVG.

The free-to-use non-commercial version doesn't require a prior written agreement with DVG but such customers, organizations and/or third parties agree by using the software and / or solution of DVG to be strongly obliged to keep all rights to this software, documentation and logotypes of the TRIP product family absolutely un infringed and protected.

Table of Contents

About This Document.....	10
Version 8.4-1.....	10
New Features in v8.4-1.....	10
Load and index batch jobs with completion info [TRIP-6270].....	10
Database list with improved performance [TRIP-6262].....	10
Version 8.4-0.....	11
New Features in v8.4-0.....	11
Access tokens [TRIP-5881].....	11
List defined views [TRIP-5873].....	11
Version 8.3-1.....	11
New Features in v8.3-1.....	11
Fetch block size for TdbFieldTermFacet [TRIP-5939].....	11
Version 8.3-0.....	11
New Features in v8.3-0.....	11
CONTROL object caching [TRIP-5837].....	11
Session-specific API/XPI logging [TRIP-5819].....	12
Corrected in v8.3-0.....	12
TdbControlCacheManager may fail with TRIPsystem 8.3+ [TRIP-5914].....	12
Version 8.2-2.....	12
New Features in v8.2-2.....	12
History deltas for FIND SAVE and RUN [TRIP-5811].....	12
Version 8.2-1.....	12
New Features in v8.2-1.....	12
Improvements for adding a large number of values [TRIP-5769].....	12
Version 8.2-0.....	13
New Features in v8.2-0.....	13
Improved session encryption [TRIP-5367].....	13
Corrected in v8.2-0.....	13
Method TdbUser.Get() fails for current user [TRIP-5041].....	13
Version 8.1-3.....	13
New Features in v8.1-3.....	13
TCP KeepAlive signaling [TRIP-5644].....	13
Version 8.1-2.....	13
New Features in v8.1-2.....	13
SortMerge property [TRIP-5590].....	13
Corrected in v8.1-2.....	14
Encrypted sessions do not work [TRIP-5562].....	14
Version 8.1-1.....	14

Corrected in v8.1-1.....	14
32-bit limitation in TdbIntegerField and TdbNumberField [TRIP-5539].....	14
Version 8.1-0.....	14
New Features in v8.1-0.....	14
Default values for database design creation [TRIP-5430].....	14
Corrected in v8.1-0.....	14
Formats with the NOLF filter not properly handled via XPI [TRIP-5488].....	14
New login attempt fails after first one failed [TRIP-5443].....	15
Version 8.0-8.....	15
Corrected in v8.0-8.....	15
Record commit may not detect deleted field contents [TRIP-5280].....	15
Version 8.0-7.....	15
Corrected in v8.0-7.....	15
Facet sorting does not always work [TRIP-5311].....	15
Version 8.0-6.....	16
News in v8.0-6.....	16
Property for overriding default facet timeout [TRIP-5310].....	16
Corrected in v8.0-6.....	16
Error encrypting passwords and tickets [TRIP-5309].....	16
Version 8.0-4.....	16
New Features in v8.0-4.....	16
Ticket subsystem status property [TRIP-5293].....	16
Ticket revocation method [TRIP-5292].....	16
Check if a record has uncommitted changes [TRIP-5287].....	16
Corrected in v8.0-4.....	16
Graph edge records not retrievable [TRIP-5288].....	16
Version 8.0-2.....	17
New Features in v8.0-2.....	17
Info on current thesaurus [TRIP-5267].....	17
Version 8.0-1.....	17
New Features in v8.0-1.....	17
Access to facet baseline update count [TRIP-5261].....	17
Databases for data from TRIPcof import connectors [TRIP-5247].....	17
Version 8.0-0.....	17
New Features in v8.0-0.....	17
Facet baseline reset [TRIP-5234].....	17
Facet baselines with TRIP kernel support [TRIP-5219].....	17
Formatted Values [TRIP-5203].....	18
Indicator for partial values [TRIP-5191].....	18
Login tickets [TRIP-5182].....	18
Field groups [TRIP-5135].....	18
XML database storage space reduction [TRIP-5134].....	19
JSON storage support for TdbXmlRecord [TRIP-5012].....	19

64-bit record and hit counts [TRIP-5004].....	19
Corrected in v8.0-0.....	19
Unnecessary network I/O in TdbSearch [TRIP-5235].....	19
Version 7.2-1.....	19
New Features in v7.2-1.....	19
Facet Baselines [TRIP-5129].....	19
Database name for TdbHistoryDetail and TdbSearchSet [TRIP-5128].....	20
Corrected in v7.2-1.....	20
Risk for connection failure with large facets.....	20
Version 7.2-0.....	20
New Features in v7.2-0.....	20
Save part of search history as procedure [TRIP-5111].....	20
DB designs from TdbDatabaseClusterDesign [TRIP-5110].....	20
New field fetch types.....	21
HTML highlighting support for local use.....	21
Support for client-side use of TRIPcof/TRIPviewX.....	21
Corrected in v7.2-0.....	21
TRIPnpx could hang indefinitely on server-side crash [NXP-98].....	21
Risk for array bounds violation in client-side text extraction and HTML conversion [NXP-100].....	21
Setting part id to retrieve a single part did not work [NXP-102].....	21
FormatException thrown from graph path analysis [NXP-103].....	21
Access violation exception if not logged in to session [NXP-104].....	21
Array bounds exception thrown if a TdbSearchSet contained a deleted record [NXP-105].....	22
Version 3.1-0.....	22
New Features in v3.1-0.....	22
Graph search and navigation [NXP-90].....	22
TdbSession.isConnected [NXP-96].....	22
Version 3.0-7.....	22
New Features in v3.0-7.....	22
Copy method for TdbRecord [NXP-93].....	22
Corrected in v3.0-7.....	23
Committing records from a TdbLocalSession fails [NXP-94].....	23
Time value may fail to parse into a DateTime [NXP-95].....	23
Version 3.0-6.....	23
New Features in v3.0-6.....	23
Support for pessimistic record locking [NXP-91].....	23
Corrected in v3.0-6.....	23
Initialization of local TRIPcof use fails [NXP-92].....	23
Version 3.0-5.....	23
Corrected in v3.0-5.....	23
Use of TdbPoolInitializer could cause crash [NXP-88].....	23
Version 3.0-4.....	24

New Features in v3.0-4.....	24
Activity logging for connection pools [NXP-85].....	24
Overloaded Acquire() method with timeout / blocking behavior [NXP-86].....	24
Session pool initialization [NXP-87].....	24
Version 3.0-3.....	24
New Features in v3.0-3.....	24
Support the TRIPview-C product [NXP-80].....	24
NOBAF option for TdbDatabaseList [NXP-84].....	24
Corrected in v3.0-3.....	25
TdbControlObject.IsFileManager returned incorrect value [NXP-81].....	25
TdbControlObject.IsUserManager returned incorrect value [NXP-82].....	25
Documentation incorrect for TdbControlObject.ManagerRights [NXP-83].....	25
Version 3.0-2.....	25
New Features in v3.0-2.....	25
TdbTextExtractionInfo.ExtractFromStored property added [NXP-72].....	25
Use TdbStringField for text extraction via TextExtractionInfo [NXP-78].....	25
Corrected in v3.0-2.....	26
Invalid characters from text extraction not handled [NXP-69].....	26
Text extraction using TdbStringField may fail [NXP-70].....	26
Text extraction update to named part records fails [NXP-71].....	26
TdbLocalSession constructor throws an exception [NXP-73].....	27
TdbLocalSession unusable with 32-bit TRIPsystem on 64-bit OS [NXP-74].....	27
Using TextExtractionInfo without setting Stream may fail [NXP-75].....	27
The TdbTimeField.Type property returns incorrect value [NXP-76].....	27
Incorrect version number check in TdbSession.SupportsFeature [NXP-79].....	27
Version 3.0-1.....	27
New Features in v3.0-1.....	27
New method TdbSearchSet.Delete() [NXP-65].....	27
Use TRIPview for HTML conversion and text extraction [NXP-67].....	28
Keyword and metadata analysis [NXP-68].....	28
Corrected in v3.0-1.....	28
Fetching large STRING values from part records [NXP-66].....	28
Version 3.0-0.....	28
News in v3.0-0.....	28
Client-side text extraction [NXP-42].....	28
Client-side HTML conversion [NXP-43].....	29
TdbControlObjectList.Contains methods added [NXP-44].....	29
Session encryption [NXP-54].....	29
On-demand term lists [NXP-55].....	29
Facet classes [NXP-56].....	29
Access to database file location names [NXP-57].....	30
Retrieve control object for a single database [NXP-60].....	30
Synchronous indexing of databases [NXP-63].....	30
Synchronous indexing of single records [NXP-64].....	30
Corrected in v3.0-0.....	30

Database language information missing from TdbDatabaseDesign [NXP-58].....	30
Chinese segmentation type "Word" not exposed [NXP-59].....	30
Version 2.1-5.....	31
Corrected in v2.1-5.....	31
String field values may get deleted on update [NXP-61].....	31
TdbDateform throws exception on partial TRIP dates [NXP-62].....	31
Version 2.1-4.....	31
New Features in v2.1-4.....	31
Improve STRING field upload performance [NXP-53].....	31
Version 2.1-3.....	31
New Features in v2.1-3.....	31
Improve STRING field download performance [NXP-51].....	31
Corrected in v2.1-3.....	32
TdbException.DebugMessage not complete [NXP-52].....	32
Version 2.1-2.....	32
News in v2.1-2.....	32
TdbDatabaseDesign.OwnerName property added [NXP-46].....	32
Filename field name for HTML conversion [NXP-49].....	32
Access rights via group memberships from TdbUserAccessList [NXP-50].....	32
Corrected in v2.1-2.....	32
Invalid exception from TdbSearchSet.ClearCache() [NXP-47].....	32
Server-side crash on Linux may hang TRIPnpx [NXP-48].....	32
Version 2.1-1.....	33
News in v2.1-1.....	33
Support for search comforters [NXP-41].....	33
Corrected in v2.1-1.....	33
No output to activity log if server exits prematurely [NXP-40].....	33
Version 2.1-0.....	33
News in v2.1-0.....	33
Open up for segmentation type "Word" [NXP-36].....	33
XPath as Query Language [NXP-37].....	33
XML Fragment Retrieval [NXP-38].....	33
Version 2.0-4.....	34
Corrected in v2.0-4.....	34
Changing password for another user does not work [NXP-34].....	34
Changing password for current user does not work [NXP-35].....	34
Version 2.0-3.....	34
News in v2.0-3.....	34
Control object method for checking if a database is XML enabled [R1459].....	34
Enable custom ASE calls for insert and update of records [R1458].....	34
Corrected in v2.0-3.....	34
Whitespace ignored between consecutive hit terms [B1433].....	34
Version 2.0-2.....	35

News in v2.0-2.....	35
Automatically close idle sessions in connection pool [R1415].....	35
Version 2.0-1.....	35
News in v2.0-1.....	35
Finer control when specifying what to fetch from a TEXT field [R1412].....	35
Version 2.0-0.....	35
News in v2.0-0.....	35
Method TdbDatabaseClusterDesign.Remove() added [R1401].....	35
Get field value size [R1373].....	35
Access field template collection via TdbRecord [R1343].....	36
New class: TdbSearch [R1342].....	36
Microsoft .NET Framework 3.5 required [R1341].....	36
FieldCount property for TdbDatabaseDesign [R1340].....	36
Improved support for RTF output from kernel windows [R1339].....	36
Generics for exposed collections [R1334].....	36
Support for thesaurus-based searching in TdbRecordSet [R1333].....	38
Access record timestamp through TdbRecord [R1317].....	38
Record ID available for new records [R1305].....	38
TdbDateform capable of formatting TRIP time stamps [R1301].....	38
Session isolation for connection pools [R1290].....	38
Support for TRIPxml 2.x added [R1124].....	38
Corrected in v2.0-0.....	38
Base64 routines produce invalid encoding [B1295].....	38
Exception thrown when saving database cluster design changes [B1400].....	39
Version 1.3-4.....	39
News in v1.3-4.....	39
Access to the did-you-mean facility [R1258].....	39
Version 1.3-3.....	39
Corrected in v1.3-3.....	39
Exception sometimes thrown in TdbDatabaseAccess.Get() for empty scopes [B1242].....	39
Version 1.3-2.....	39
News in v1.3-2.....	39
TRIPgrid added to TdbProduct enum [R1199].....	39
Corrected in v1.3-2.....	39
Record name not parsed for databases not owned [B1213].....	39
Version 1.3-1.....	40
Corrected in v1.3-1.....	40
Removal of all values from a field not working [B1183].....	40
Version 1.3-0.....	40
News in v1.3-0.....	40
Tuple lists [R1151].....	40
Value count property for TdbField [R1153].....	40
Value insert method for TdbField [R1154].....	40

Value remove method for TdbField [R1155].....	40
Date format conversion class [R1169].....	40
Corrected in v1.3-0.....	41
Setting a numeric field to an empty string caused exception [B1156].....	41
Exception thrown if decimal separator in locale is not a dot [B1157].....	41
Version 1.2-2.....	41
News in v1.2-2.....	41
Unicode session awareness [R1145].....	41
Corrected in v1.2-2.....	41
Resource leaks in TdbStringField [B1039].....	41
Exception thrown when using TdbRecordSet without a retrieval template [B1144].....	41
Version 1.2-1.....	41
News in v1.2-1.....	41
Connection pool shutdown.....	41
Re-establishing broken pooled sessions.....	41
Version 1.2-0.....	42
News in v1.2-0.....	42
Language support for connection pools.....	42
Connection pool limit and size.....	42
Uniform casing for method and property names.....	42
TRIPview 2.0 HTML highlighting.....	42
Property for assigned classification categories.....	42
Reverse retrieval of record sets.....	42
Corrected in v1.2-0.....	42
Exception thrown when using unnamed part records.....	42
TRIPnpx assembly not marked as CLS-compliant.....	42
Version 1.1-0.....	43
News in v1.1-0.....	43
New namespace name.....	43
Corrected in v1.1-0.....	43
Connection pools not working.....	43
Known Remaining Issues.....	43

About This Document

This document contains the change history for TRIPnpx.

Version 8.4-1

New Features in v8.4-1

Load and index batch jobs with completion info [TRIP-6270]

The TdbDatabaseDesign class now provides the method SubmitBatchJob, which can be used to submit LOAD and INDEX requests in a manner that returns a job ID for the submitted job.

Completion messages are available via the TdbSession class, which provides the new property JobResultSink that is used to register a callback object of a custom class that implements the new TdbJobResultSink interface. Details about completed jobs, including their IDs, will be sent to this object via its CompletedJob method.

Completion information can be received with any call to TRIPnpx that results in network I/O. If no such calls can reasonably be made, the TdbSession now also provides the new pollJobResults method whose sole purpose is to check for completed batch jobs.

NOTE that other batch jobs such as PRINT and UPDATE are not supported for this type of job completion feedback, nor are any batch jobs submitted via CCL commands.

The example program BulkPut has been extended to illustrate how this new behavior can be used.

This feature requires TRIPsystem 8.4-6 or later.

Database list with improved performance [TRIP-6262]

The TdbDatabaseList class now provides a new constructor that takes a TdbDatabaseListType enumerated value as its third parameter. If this parameter is set to DatabaseListNoCounts, the list will be produced noticeably faster. The caveat is that no record counts will be provided for this type of list.

A new example program DatabaseList is provided to illustrate how this new behavior can be used.

This feature requires TRIPsystem 8.4-6 or later.

Version 8.4-0

New Features in v8.4-0

Access tokens [TRIP-5881]

The TdbSession class now supports using access and refresh tokens for accessing TRIPsystem. Users of the ticket mechanism should consider migrating to tokens.

To use this feature, it must first be enabled in TRIPsystem. Please refer to TRIPsystem documentation for more information.

This feature requires TRIPsystem 8.4-0 or later.

List defined views [TRIP-5873]

A new method GetDefinedViews() has been added to the TdbSession class. It returns a dictionary over the views defined in the current session. Such views are defined using the "DEFINE VIEW" CCL command, either explicitly from the application or implicitly in start procedures or via field group declarations in database designs.

This feature requires TRIPsystem 8.4-0 or later.

Version 8.3-1

New Features in v8.3-1

Fetch block size for TdbFieldTermFacet [TRIP-5939]

The new property FetchSize for the TdbFieldTermFacet class can be used to override the default number (100) of facet values to retrieve from TRIPsystem per network request.

Set this property to a higher value if there are more than 100 values and your application need immediate access to all of them.

Version 8.3-0

New Features in v8.3-0

CONTROL object caching [TRIP-5837]

A new class TdbControlCacheManager has been added, accessible via the TdbSession property ControlCache. This new class provides an efficient means by which CONTROL information such as database lists, database designs, etc, can be obtained with a minimum of overhead. Applications are strongly recommended to be migrated to use this new feature.

When TRIPsystem 8.3-0 or later is used, the TdbControlCacheManager will in addition be able to preload data faster than with older versions of TRIPsystem.

Session-specific API/XPI logging [TRIP-5819]

Session-specific server-side logging can now be enabled and accessed via new methods of the TdbSession class. Logs possible to enable and access this way are the TRIP kernel API log and the server-side XPI log.

Enabling and obtaining logs this way is recommended especially for production servers.

This feature requires TRIPsystem 8.3-0 or later.

Corrected in v8.3-0

TdbControlCacheManager may fail with TRIPsystem 8.3+ [TRIP-5914]

The TdbControlCacheManager class that was introduced in version 8.3-0:0 of TRIPnpx could throw an exception when used with TRIPsystem of version 8.3-0 and later.

Corrected in 8.3-0:1

Version 8.2-2

New Features in v8.2-2

History deltas for FIND SAVE and RUN [TRIP-5811]

The TdbCclCommand and TdbSearch classes have a new property [get|set]HistoryDelta. If this is set to true, all procedure execution commands (FIND SAVE and RUN) returns search history updates for all sets generated by the procedure. History updates are represented by the TdbHistoryDetail class when using TdbCclCommand, and by the TdbSearchSet class when using TdbSearch.

Note that when using the RUN command to execute procedures that contain the RENUM, LIST, CLOSE or DELETE SEARCH commands, a full history replacement will be returned even if the history delta behavior is otherwise enabled.

If history deltas are disabled (the default and backward-compatible behavior), the search history update will for FIND SAVE only contain the last set generated by the procedure, and for RUN it will be complete replacement of the entire search history.

Requires TRIPsystem 8.2-3 or later.

Version 8.2-1

New Features in v8.2-1

Improvements for adding a large number of values [TRIP-5769]

The TdbField classes have two new methods; AppendValues and InsertValues. These should be used when you have a lot of values (subfields or paragraphs) to add per field in a record. Using these will yield significant performance improvements compared to using the regular AppendValue and InsertValue to add one value at a time.

In order to get best performance from also the commit operation, use of TRIPsystem 8.2-1:3 or later is strongly recommended.

See also the new sample program BulkPut for a demonstration of the use of the AppendValues method and a comparison with the regular AppendValue method.

Version 8.2-0

New Features in v8.2-0

Improved session encryption [TRIP-5367]

Encrypted sessions against TRIPsystem 8.2 and later will now use session-specific AES keys for encryption, resulting in enhanced security and improved performance.

Corrected in v8.2-0

Method TdbUser.Get() fails for current user [TRIP-5041]

Calling the get() method of the TdbUser class without having created the TdbUser instance via a TdbUserList or previously fetched a user's profile with the same instance should retrieve the currently logged on user's profile. This always failed with a TdbException containing the message "Invalid arguments passed to function".

Corrected in 8.2-0:2.

Version 8.1-3

New Features in v8.1-3

TCP KeepAlive signaling [TRIP-5644]

TRIPnpx will now use TCP KeepAlive signals for all sessions. The values for the time until start signaling, the interval between signals and after how many dropped are determined on an OS level. These normally have the default values 7200000 (2 hours), 1000 (1 second) and 5, respectively.

The keepalive time and interval values can also be set on a per-session basis using a new override for the constructor of the TdbTripNetSession class.

When using a proxy or load balacer between TRIPnpx and TRIPsystem, lowering the keepalive time from its 2 hour default may be needed to protect against connections being dropped after a few minutes of idling.

Version 8.1-2

New Features in v8.1-2

SortMerge property [TRIP-5590]

A new property SortMerge has been added to the TdbRecordSet and TdbSearchSet classes. This should always be used by applications to control sort merge behavior for cluster search results instead of relying on TRIPsystem default.

This feature requires TRIPsystem version 8.1-1:3 or later.

Corrected in v8.1-2

Encrypted sessions do not work [TRIP-5562]

Any attempt to establish an encrypted session failed with an "unexpected state" exception.

Corrected in 8.1-2:1.

Version 8.1-1

Corrected in v8.1-1

32-bit limitation in TdbIntegerField and TdbNumberField [TRIP-5539]

The classes TdbIntegerField and TdbNumberField validates the values assigned to them when using the methods appendValue, insertValue and setValue. This validation incorrectly used 32-bit validation even when connected to a TRIPsystem installation capable of handling 64-bit numerical values.

Version 8.1-0

New Features in v8.1-0

Default values for database design creation [TRIP-5430]

A new overloaded version of the Get method of the TdbDatabaseDesign class has been implemented. This takes two argument; a string with the name of the database whose design to retrieve, and a boolean that indicates if the design is retrieved for subsequent creation or update (true) or for reference/inspection only (false).

This new overloaded Get method should be called with the second parameter set to true as the first call after having created a new TdbDatabaseDesign instance with the intention of using it for the creation or alteration of a database design. This populates the design instance with TRIPsystem-assigned default values when the database does not previously exist, thereby reducing the necessity to explicitly specify some of the mandatory parameters such as database file paths.

This feature requires TRIPsystem version 8.1-0 or later.

Corrected in v8.1-0

Formats with the NOLF filter not properly handled via XPI [TRIP-5488]

Reports using an output format with the NOLF filter will be rendered without TRIPsystem doing any sort of line planning. This did not have any effect for SHOW orders issued by TRIPnpx.

The TdbKernelWindow class will now keep track of the NOLF filter use in the retrieved window buffer. The ToString() method will not apply linebreaks for output format reports generated using the NOLF filter. For applications that wish to render their own output, the line feed status for the current window buffer is available through the new LinePlanned property.

Note that for SHOW orders from clusters with mixed output formats where some databases use a NOLF filter and some do not, the linefeed behavior is undefined for window buffers that contain information from more than one database.

This correction requires TRIPsystem version 8.1-0:4 or later.

New login attempt fails after first one failed [TRIP-5443]

If a login attempt has failed (e.g. due to incorrect password), the session could immediately become unusable. The intended behavior is that secondary login attempt should be possible to make without having to create a new session object and try again with that.

Version 8.0-8

Corrected in v8.0-8

Record commit may not detect deleted field contents [TRIP-5280]

Removing all values from a field would cause TRIP not to update the field upon commit if the application implemented the following sequence:

- A new TdbRecord instance is created and populated with values.
- The TdbRecord instance is committed, creating a new record in TRIP.
- Using the same TdbRecord instance, a field is cleared from all values.
- The TdbRecord instance is committed again.

Version 8.0-7

Corrected in v8.0-7

Facet sorting does not always work [TRIP-5311]

Requesting a facet to be sorted by text descending or frequency ascending would instead yield the default sorting direction of text ascending and frequency descending, respectively.

The correction of this issue also involves changes to TRIPsystem. To ensure that this issue is not encountered, make sure to use TRIPsystem 8.0-7 or newer.

Version 8.0-6

News in v8.0-6

Property for overriding default facet timeout [TRIP-5310]

The TdbFacet.DefaultTimeout property has been added. This static property can be used to override the default facet timeout (1800 seconds) so that the TdbFacet.Timeout property does not have to be set for every facet request.

Corrected in v8.0-6

Error encrypting passwords and tickets [TRIP-5309]

In rare cases a user would not be able to log in even though the password was correctly specified. The same problem could also occur when logging in using a ticket instead of username and password.

The correction of this issue also involves changes to TRIPsystem. To ensure that this issue is not encountered, make sure to use TRIPsystem 8.0-6 or newer.

Version 8.0-4

New Features in v8.0-4

Ticket subsystem status property [TRIP-5293]

The TdbSession.TicketsEnabled property has been added. It returns true if the login ticket subsystem is enabled and the RequestLoginTicket() method can be expected to succeed.

This feature requires TRIPsystem version 8.0-4 or later.

Ticket revocation method [TRIP-5292]

The TdbSession.RevokeLoginTicket() method has been added. Use it to revoke the login ticket for the current user, or when logged in as SYSTEM to revoke the ticket for another user.

This feature requires TRIPsystem version 8.0-4 or later.

Check if a record has uncommitted changes [TRIP-5287]

The TdbField and TdbRecord classes have been extended with a new property Changed that returns true if the field or record contains uncommitted changes.

Corrected in v8.0-4

Graph edge records not retrievable [TRIP-5288]

Attempting to retrieve graph edge records from a TdbGraphSet instance would return an error message like "The record at index NNN is not a graph record."

The fix for this problem involves changes in both TRIPnpx and TRIPsystem. The minimum required version of TRIPsystem for use of graph functionality is therefore 8.0-4.

Version 8.0-2

New Features in v8.0-2

Info on current thesaurus [TRIP-5267]

The name and definition of the currently defined thesaurus is now available via the properties `CurrentThesaurusName` and `CurrentThesaurusDef` in the `TdbCclCommand` and `TdbSearch` classes. This thesaurus information is available after a CCL command has been executed via the related `TdbSearch` or `TdbCclCommand` instance, and that command has involved a thesaurus definition or the generation of a search set.

This feature requires TRIPsystem version 8.0-2 or later.

Version 8.0-1

New Features in v8.0-1

Access to facet baseline update count [TRIP-5261]

The count of values with a non-zero record count for facets that use baselines is now available via the `UpdateValueCount` property of the `TdbFacet` class.

This feature requires TRIPsystem version 8.0-1 or later. For older versions of TRIPsystem, and for facets that do not use baselines, the `UpdateValueCount` property will always contain the same value as the `ValueCount` property.

Databases for data from TRIPcof import connectors [TRIP-5247]

Support for TRIPcof import connector databases has been added. Such databases have a pre-defined design required by TRIPcof import connectors and can be created by setting the `IsConnectorDatabase` property of the `TdbDatabaseDesign` class to true.

This feature requires TRIPsystem version 8.0-1 or later.

Version 8.0-0

New Features in v8.0-0

Facet baseline reset [TRIP-5234]

This feature requires TRIPsystem version 8.0-0 or later.

The new `TdbFacet.ResetBaseline` property will, if assigned to true, cause the most recent record counts for the specified facet to be copied to the baseline, overwriting the previous baseline counts. This operation will keep the previous facet values, even values that have a baseline count of zero.

See the updated example program `FacetBaseline` for an example of how to use this functionality.

Facet baselines with TRIP kernel support [TRIP-5219]

This feature requires TRIPsystem version 8.0-0 or later.

Facets that use baselines will with TRIPsystem 8.0 and later benefit from added performance and added features such as the option to have baseline values inline with the regular values, access of baselined facets without needing to perform a refinement, and access to any facet value without having to explicitly fetch ranges of values from the server.

The properties `TdbFacet.FetchFrom`, `TdbFacet.FetchTo` and `TdbFacet.BaselineSize` have been DEPRECATED. Please refer to the reference manual for information on how to modify your application.

The example applications `Facet` and `FacetBaseline` have been updated to illustrate the use of facets with and without baselines with version 8 of TRIPnpx and TRIPsystem.

The facet chapter in the "TRIPnpx & TRIPjxp Programmer's Guide" has been updated. Note especially the new example of how to retrieve an existing facet more efficiently (e.g. when displaying more values in response to a scrolling operation).

Formatted Values [TRIP-5203]

This feature requires TRIPsystem version 8.0-0 or later.

The new `TdbFieldFetchType.FormattedValue` enum can be used to request the result from an output format to be returned as if it were a text field value. Use with the `TdbFieldTemplate` class in search set retrieval templates.

Indicator for partial values [TRIP-5191]

This feature requires TRIPsystem version 8.0-0 or later.

The `TdbField.Partial` property has been added. This property is true if the retrieved value held by the `TdbField` instance is partial. Partial values are typically produced when requesting hit-focused, part-merged fragments from larger text or phrase fields.

Login tickets [TRIP-5182]

This feature requires TRIPsystem version 8.0-0 or later.

The `TdbSession` class now can be used to request a login ticket for a logged in user. This ticket can in subsequent logins be used instead of username and password to enable an SSO-like behavior in applications.

To use this feature, it must first be enabled in TRIPsystem. Please refer to TRIPsystem documentation for more information.

Field groups [TRIP-5135]

This feature requires TRIPsystem version 8.0-0 or later.

A new class `TdbFieldGroupDesign` has been added. It is used together with the `TdbDatabaseDesign` class in order to administer field groups, a new feature supported from version 8.0 of TRIP.

The `TdbTupleList` class has been given an additional constructor that can initialize a tuple list using a field group.

XML database storage space reduction [TRIP-5134]

TRIPsystem 8.0 introduced a new design for JSON/XML databases that uses less disk space than the previous design. As part of this overhaul, the default value for the StoreCopy property of the TdbXmlRecord class has been changed from true to false. If your application stores XML documents and still need to have binary copies included, make sure to assign set StoreCopy to true before committing the record.

JSON storage support for TdbXmlRecord [TRIP-5012]

This feature requires TRIPsystem version 8.0-0 or later.

The TdbXmlRecord class has been adapted to the JSON storage functionality in TRIPsystem 8.0. Both XML and JSON documents can be stored in XML databases, and retrieved as either JSON or XML regardless of the original format. XPath can also be used to search in documents whose original format is JSON.

64-bit record and hit counts [TRIP-5004]

This feature requires TRIPsystem 8.0-0 or later.

Record counts and hit counts for search sets and term lists are now available as 64-bit integer (long) values. New properties and methods for this purpose have been added to the classes TdbHistoryDetail, TdbTerm, TdbTermList, TdbRecordSet, TdbSearchSet and TdbFacetTermList.

Accessing a record count or hit count larger than Int32.MaxValue (2147483647) via the older 32-bit properties will cause the value to be truncated to 2147483647.

Only applications that work with a very large number of records or need to operate on very large hit counts need to use the new 64-bit record count and hit count properties.

Corrected in v8.0-0

Unnecessary network I/O in TdbSearch [TRIP-5235]

The automatic retrieval template functionality of the TdbSearch class allows data to be retrieved by an application without explicitly having to specify the fields to retrieve. This template was always created upon successful completion of a CCL order that results in a search set, even if the application ends up not retrieve any data from the search set. This caused some unnecessary network I/O and slightly longer response times.

This behavior has been changed so that the automatic retrieval template, if enabled, is not created until it is known that a template will be needed. This change will result in a minor performance improvement.

Version 7.2-1

New Features in v7.2-1

Facet Baselines [TRIP-5129]

This feature requires TRIPsystem 7.2-1 or later.

A facet baseline enables the recall of a previous, initial set of facet values so that a new facet request can 1) include the previous counts as a comparison, and 2) add values from the baseline to the new result if the new result is smaller than the requested range.

A new example program FacetBaseline has been added to exemplify the usage of this new feature.

Database name for TdbHistoryDetail and TdbSearchSet [TRIP-5128]

This feature requires TRIPsystem 7.2-1 or later.

The name of the database or cluster that was used to produce a search set is now available via the TdbHistoryDetail and TdbSearchSet classes.

The example program Search has been modified to also show this information.

Corrected in v7.2-1

Risk for connection failure with large facets

When using the TdbFacet classes to generate facet values such that the total number of values for at least one facet exceeded 100, an incorrect request was sent by TRIPnpx to TRIPsystem that caused an instability that would result in a server-side crash a couple of minutes later.

Version 7.2-0

New Features in v7.2-0

Save part of search history as procedure [TRIP-5111]

The TdbProcedure class has been extended with a method called SaveSearch, that saves the CCL commands associated with a specified search set as a procedure, complete with dependent commands such as BASE and renumbered search set references (S=<num>).

This feature requires TRIPsystem 7.2 or later.

DB designs from TdbDatabaseClusterDesign [TRIP-5110]

The TdbDatabaseClusterDesign class can now provide a list of all database designs for databases in a cluster, with nested clusters resolved. Use the new method GetMemberDesigns() for this.

The TdbDatabaseClusterDesign.Get() method can now be asked to resolve nested clusters into their constituent databases. Note that the TdbDatabaseClusterDesign instance will not support updating the cluster after this option has been used.

The new method GetCommonFields() of the TdbDatabaseClusterDesign class provides access to a list of TdbFieldDesign instances that represent the fields common to all databases in the cluster.

These new features require TRIPsystem 7.2 or later.

New field fetch types

New field fetch types have been added to the TdbFieldFetchType enum. When used, these new types provide additional control over what parts of a field of a record in a search set to retrieve.

HTML highlighting support for local use

Extended the HTML conversion functionality of the TdbStringField class to support hit highlighting when a local TRIPcof or TRIPviewX installation is used.

Support for client-side use of TRIPcof/TRIPviewX

A local installation of TRIPviewX/TRIPcof can now be used in client-side mode when performing text extraction and HTML conversion. The sample ExtractText has been extended to illustrate client-side use. A new sample HTMLConvert has been added to illustrate use of HTML conversion.

Corrected in v7.2-0

TRIPnpx could hang indefinitely on server-side crash [NXP-98]

If the tbserver process crashes or is killed, there was a risk that TRIPnpx would hang indefinitely waiting for response data from the server.

Risk for array bounds violation in client-side text extraction and HTML conversion [NXP-100]

There was a risk for an array bounds violation in the handling of response data from the client-side text extraction and HTML conversion routines.

Setting part id to retrieve a single part did not work [NXP-102]

Setting the PartId property of the TdbRecord class in order to retrieve a single part record had no effect.

FormatException thrown from graph path analysis [NXP-103]

A FormatException could be thrown from the graph path analysis methods if the local decimal separator happened to be something else than a dot.

Access violation exception if not logged in to session [NXP-104]

If a TdbSession object has been created but the Login() method has not been successfully called, TRIP will crash on an access violation if TRIPnpx is nevertheless being used.

TRIPnpx will now prior to any call that is not a login/authentication check the state of the session and throw a TdbException with code TdbException.NOT_LOGGED_IN if the login method has not been successfully called.

Array bounds exception thrown if a TdbSearchSet contained a deleted record [NXP-105]

If a record is deleted using the `TdbRecord.Delete()` method and a search immediately thereafter is conducted via the `TdbSearch` class such that the resulting search set contains the record, then an array bounds exception would be thrown when fetching records from the set using `TdbSearchSet`.

This error would only occur if the database was not indexed between the delete and the search operations.

As part of the correction for this issue a new property 'Deleted' has been added to the `TdbRecord` class. This property will return true if the object represents a deleted record and the session is connected to TRIPsystem version 7.2 or later.

Version 3.1-0

New Features in v3.1-0

Graph search and navigation [NXP-90]

A namespace `TietoEnator.Trip.Nxp.Graph` has been added. It contains APIs for the use of graph databases in TRIP. For more information, please refer to the Programmer's Guide.

This feature requires TRIPsystem 7.1 or later.

Graph support is EXPERIMENTAL in version 7.1 of TRIPsystem and version 3.1 of TRIPnxp. Its use is encouraged and Tieto would like feedback on it. This feedback will help determine the future direction of the development of graph database support in TRIP.

TdbSession.IsConnected [NXP-96]

A property `IsConnected` has been added to the `TdbSession` class. This reports true if the session is connected to the server via a socket or if the session is web or local. False is returned for networked sessions that no longer have an active socket connection to the TRIP server.

This method is useful if TRIP access code is implemented in object destructors that may be executed by the runtime after the session has been terminated.

Version 3.0-7

New Features in v3.0-7

Copy method for TdbRecord [NXP-93]

A new method "Copy" for record copying has been added to the `TdbRecord` class. It is strongly recommended that all existing record copying routines in TRIPnxp applications are modified to use this new method if possible. This is because it will offer better performance, plus that it fully supports STRING fields, values of which may silently be omitted from the copy when using TRIPnxp version 2.1-3 or later.

This functionality requires TRIPsystem 6.2-11 or later or TRIPsystem 7.0-4 or later in order to work.

Corrected in v3.0-7

Committing records from a TdbLocalSession fails [NXP-94]

Committing a new or modified record in a TRIPnpx application that uses a TdbLocalSession to connect to TRIP would throw a NullPointerException.

Time value may fail to parse into a DateTime [NXP-95]

The FormatTime(String) method of the TdbDateform class was unable to parse time values from TRIP where the hour, minute or second value was lower than 10.

Version 3.0-6

New Features in v3.0-6

Support for pessimistic record locking [NXP-91]

Methods Lock() and Unlock() have been added to the TdbRecord class in order to support pessimistic (exclusive) record locking. This kind of locking differs from the default optimistic locking in that it blocks other user's attempts to update the record until the lock is released.

This feature requires TRIPsystem 7.0-3 or later.

Corrected in v3.0-6

Initialization of local TRIPcof use fails [NXP-92]

TRIPnpx attempted to load the TRIPcof file filter configuration file using an invalid path. This resulted in TRIPcof being impossible to use locally from TRIPnpx.

Version 3.0-5

Corrected in v3.0-5

Use of TdbPoolInitializer could cause crash [NXP-88]

There was a risk for a TdbException to be thrown when the TdbPoolInitializer was used for the execution of TRIP statements that will apply to the entire lifetime of a pooled session.

This problem was also corrected in TRIPnpx 2.1 as per the release of TRIPnpx 2.1-10.

Version 3.0-4

New Features in v3.0-4

Activity logging for connection pools [NXP-85]

Activity logging is now supported in the TdbConnectionPool class. Assign a TdbActivityLogger instance to a pool using the new ActivityLog property. This will log the activity of the pool itself for problem analysis purposes.

This feature was also added to TRIPnpx 2.1 as per the release of TRIPnpx 2.1-9.

Overloaded Acquire() method with timeout / blocking behavior [NXP-86]

The TdbConnectionPool class has been given a new overloaded version of the Acquire() method. It takes a long integer as argument that denotes a period in milliseconds to wait until a session becomes available if one is not available immediately.

This feature was also added to TRIPnpx 2.1 as per the release of TRIPnpx 2.1-9.

Session pool initialization [NXP-87]

The TdbPoolInitializer delegate has been added to TRIPnpx. When the new PoolInitializer property of the TdbConnectionPool class is assigned a reference conforming to the TdbPoolInitializer delegate, the referenced method will be called for each session added to the pool.

The use of this new property is strongly recommended in high-load situations when several operations need to be executed (e.g. opening databases, executing defines, etc) prior to the actual operation the application requires.

This feature was also added to TRIPnpx 2.1 as per the release of TRIPnpx 2.1-9.

Version 3.0-3

New Features in v3.0-3

Support the TRIPview-C product [NXP-80]

TRIPnpx can now detect the TRIPview-C product in the license of the currently connected TRIPsystem installation. The TdbProduct enumeration has been modified accordingly and documentation that refer to TRIPview has been modified when it also applies to TRIPview-C.

NOBAF option for TdbDatabaseList [NXP-84]

Using the TdbDatabaseList class to generate a list of databases now supports a NOBAF parameter to the constructors. Setting this to true causes the list to be produced without any access to the BAF files of the databases. This gives better performance in situations with a large number of databases and when the BAF files are stored on a storage medium with slower access times.

Using the NOBAF parameter set to true means that information that is solely stored in the BAF file will not be returned. This includes the record count, which instead will be reported as the last total number of indexed records (any unindexed records will not be reported). Neither will the time for the last database update be reported.

This feature requires TRIPsystem 7.0.1 or later, and databases must be reindexed with in order for the count of indexed records to be correctly reported.

Corrected in v3.0-3

TdbControlObject.IsFileManager returned incorrect value [NXP-81]

The IsFileManager property of the TdbControlObject class returned true only if the control object represents a user that is a user manager.

TdbControlObject.IsUserManager returned incorrect value [NXP-82]

The IsUserManager property of the TdbControlObject class returned true only if the control object represented a user that is a file manager.

Documentation incorrect for TdbControlObject.ManagerRights [NXP-83]

The bitmask returned by the ManagerRights property of the TdbControlObject class was incorrectly documented. It said that bit 0 indicates file management and bit 1 indicates user management, while it actually is the other way around.

Version 3.0-2

New Features in v3.0-2

TdbTextExtractionInfo.ExtractFromStored property added [NXP-72]

While it has been possible to extract text from data previously stored in a STRING field, the support for this has been depending on ambiguous use of programming interfaces and success has been context dependent.

To reduce ambiguity, the TdbTextExtractionInfo class has been extended with a new property ExtractFromStored that must be set to true if text extraction is to be done from a value previously stored in the STRING field referred to by the BinaryCopyField property. It is still also necessary to set the FileName property, but this is to inform TRIPview/TRIPcof of the file name/type. The file itself does not have to exist, nor does it get transferred to the server.

This functionality requires TRIPsystem 6.2-9:20 or later or TRIPsystem 7.0-0:4 or later in order to work.

This feature was also added to TRIPnpx 2.1 as per the release of TRIPnpx 2.1-8.

Use TdbStringField for text extraction via TextExtractionInfo [NXP-78]

It is now possible to use the data assigned to a TdbStringField instance (via the Blob property or the CopyFromFile method) in text extraction specified via TdbTextField.TextExtractionInfo.

This works so that when the new property `TdbTextExtractionInfo.StringField` is assigned a non-null value, text extraction will be performed from any value assigned to the provided `TdbStringField` instance. This works like if the file data had been provided to the `TdbTextExtractionInfo.Stream` property.

When the `TdbTextExtractionInfo.StringField` property is assigned a non-null value, any value assigned to the `TdbTextExtractionInfo.Stream` property will be ignored. The property `TdbTextExtractionInfo.FileName` is still required, though.

Note that if text extraction is to be performed from a value already stored in a string field, it is better not to include the `TdbStringField` instance at all in the `TdbRecord` object used for commit. In this case, set the property `TdbTextExtractionInfo.ExtractFromStored` to true instead.

This functionality requires TRIPsystem 6.2-9:20 or later or TRIPsystem 7.0-0:4 or later in order to work.

This feature was also added to TRIPnxp 2.1 as per the release of TRIPnxp 2.1-8.

Corrected in v3.0-2

Invalid characters from text extraction not handled [NXP-69]

Text extraction (and re-extraction when creating an HTML conversion with highlight) using TRIPview or TRIPcof may sometimes produce characters that are not valid in XML text. This will result in the text not being possible to import into TRIP.

Text extraction using `TdbStringField` may fail [NXP-70]

Text extraction from data added to a STRING field using the `ExtractionTarget` property of the `TdbStringField` class in TRIPnxp did not work if `CopyFromFile` was used to load data into the field and the `ExtractionTarget` property was set before the call to `CopyFromFile`.

The operation would fail silently without throwing an exception or otherwise indicating any error. It would appear to succeed, but no content would be extracted to the referred TEXT field.

This problem was also corrected in TRIPnxp 2.1 as per the release of TRIPnxp 2.1-8.

Text extraction update to named part records fails [NXP-71]

Text extraction (using the classes `TdbTextExtractInfo` and `TdbTextField`) during update of a record does not work if the database has a part record name field defined. The operation appeared to succeed (no error reported), but the contents of the TEXT and STRING fields would be incorrect. Depending on usage scenario, either of these fields would be empty or retain the old value.

This problem occurs with TRIPnxp versions 2.1-4 - 2.1-7, and 3.0-0 - 3.0-1.

This problem was also corrected in TRIPnxp 2.1 as per the release of TRIPnxp 2.1-8.

TdbLocalSession constructor throws an exception [NXP-73]

It was not possible to use TRIPNxp to connect to a local TRIPsystem using the TdbLocalSession class. An exception carrying the message "Illegal attempt to invoke methods on an unsuitable session" is thrown.

This problem occurs with TRIPNxp versions 2.1-3 - 2.1-7, and 3.0-0 - 3.0-1.

This problem was also corrected in TRIPNxp 2.1 as per the release of TRIPNxp 2.1-8.

TdbLocalSession unusable with 32-bit TRIPsystem on 64-bit OS [NXP-74]

It was not possible to use the TdbLocalSession class to connect to a local 32-bit TRIPsystem with the 32-bit build of TRIPNxp. This was caused by the 32-bit build actually being "Any CPU" and not strictly 32-bit.

TRIPNxp is now available in three different builds; 32-bit (x86), 64-bit (x64) and generic (Any CPU). If the TdbLocalSession class is not used, the generic Any CPU build is the recommended one.

Using TextExtractionInfo without setting Stream may fail [NXP-75]

Using TdbTextField.TextExtractionInfo to perform text extraction without setting the Stream property would not work properly against TRIPsystem versions older than 6.2-9:8

This problem occurs with TRIPNxp versions 2.1-4 - 2.1-7, and 3.0-0 - 3.0-1.

This problem was also corrected in TRIPNxp 2.1 as per the release of TRIPNxp 2.1-8.

The TdbTimeField.Type property returns incorrect value [NXP-76]

The TdbTimeField.Type property incorrectly reported TdbFieldType.TextField instead of TdbFieldType.TimeField.

This problem was also corrected in TRIPNxp 2.1 as per the release of TRIPNxp 2.1-8.

Incorrect version number check in TdbSession.SupportsFeature [NXP-79]

The TdbSession.SupportsFeature incorrectly handled patch numbers, which could result in some features being incorrectly reported as not supported.

This problem was also corrected in TRIPNxp 2.1 as per the release of TRIPNxp 2.1-8.

Version 3.0-1

New Features in v3.0-1

New method TdbSearchSet.Delete() [NXP-65]

A new instance method Delete() has been added to the TdbSearchSet class. This method should be used for the deletion of individual search sets created via a TdbSearch instance instead of explicitly issuing a DELETE CCL order.

This feature was also added to TRIPnpx 2.1 as per the release of TRIPnpx 2.1-6.

Use TRIPview for HTML conversion and text extraction [NXP-67]

The `TdbStringField.Convert()` method can now be used with a local installation of TRIPview 3.0 or later.

The client-side text extraction behavior of the `TdbRecord` class via the `TdbTextExtractionInfo` class can now be used with a local installation of TRIPview.

Keyword and metadata analysis [NXP-68]

TRIPnpx now supports keyword and metadata analysis of text in records using TRIPcof and TRIPnlp. Analysis is performed upon commit of a new or modified record if the `TdbNlpInfo` instance accessible via the `TdbRecord.NlpInfo` property has been configured.

This feature requires TRIPsystem 7.0 or later with TRIPcof installed on the server side. TRIPnlp or a custom text analysis adapter for TRIPcof is also required.

Corrected in v3.0-1

Fetching large STRING values from part records [NXP-66]

Incorrect part record information is can be supplied by TRIPsystem for the retrieval of large (>2MB) string field values. This causes STRING values for part record 2..N to always be fetched from the first part record instead.

When storing a large (>2MB) STRING field value for part record nr 2..N, the STRING value for the last part record would be assigned to all part records. This was due to missing part record information in the communications between TRIPsystem and TRIPnpx for the stream transfer of large binary values.

TRIPsystem version 6.2-9:18 or later and version 7.0-0:3 or later are also required on the server side to make part record storage work correctly.

This problem occurs in TRIPnpx versions 2.1-3 - 2.1-6 and in 3.0-0. This fix was also introduced into the 2.1 release series in version 2.1-7.

Version 3.0-0

News in v3.0-0

Client-side text extraction [NXP-42]

Added support for client-side text extraction using a local TRIPcof installation. Assign the `ClientSide` property to true on the `TdbTextExtractionInfo` instance for the text field in which to store the extracted text.

Do not pass file data using the only the `Stream` property; either pass the name of an existing local file using the `FileName` property, or pass the file data using a `Stream` and the name of the file in the `FileName` property.

This feature requires a local installation of TRIPcof.

For more information about client-side use of TRIPcof from TRIPnpx, please refer to the Programmer's Guide. Also take a look at the CofClient sample program.

Client-side HTML conversion [NXP-43]

Added support for client-side HTML conversion using a local TRIPcof installation. Use the new method `TdbStringField.Convert` to perform client-side conversion operations.

This feature requires a local installation of TRIPcof.

For more information about client-side use of TRIPcof from TRIPnpx, please refer to the Programmer's Guide. Also take a look at the CofClient sample program.

TdbControlObjectList.Contains methods added [NXP-44]

The `TdbControlObjectList` class has been enhanced with two new versions of the `Contains` method. One takes a name and a type and the other a name only. These can be used for more straight-forward checking of existence, e.g. the membership of users in groups.

Session encryption [NXP-54]

Support for encrypted sessions has been added. To establish an encrypted session, use one of the new overloaded constructors for the `TdbTripNetSession` class.

This feature requires TRIPsystem 7.0 or later to benefit from. With older versions of TRIPsystem, requesting an encrypted session will result in an error.

On-demand term lists [NXP-55]

Support for on-demand term lists has been added to the `TdbSearch` and `TdbCclCommand` classes. Terms from such term lists can now be retrieved on demand and in blocks of 100 terms each, which improves retrieval performance for large term lists. This behavior can be enabled using the `UsePagedTermLists` property (available on both the `TdbCclCommand` and `TdbSearch` classes).

Multiple on-demand term lists can also be used concurrently.

NOTE: The interface of the `TdbTermList` class has changed slightly in order to accommodate this enhancement. If you built your application using a version of TRIPnpx older than 3.0, you should recompile your application in order to check if you have to make any modifications to your code prior to deploying TRIPnpx.

This feature requires TRIPsystem 7.0 or later to benefit from. With older versions of TRIPsystem, all terms from a term lists will always be returned as a result of the `DISPLAY` command.

Facet classes [NXP-56]

A new namespace `TietoEnator.Trip.Nxp.Facet` has been added that contains classes for generation of facets based on regular field-based term lists, KVP term lists, record classification and collections of searches.

A new example program `FacetSet` is provided. It illustrates the use of all four types of facets.

This feature requires TRIPsystem 7.0 or later to benefit from.

Access to database file location names [NXP-57]

A class `TdbLocations` in the `TietoEnator.Trip.Nxp.Database` namespace has been added. This class provides access to TRIP configuration entries that map to directory paths so that the corresponding symbols can be used as location names by custom database creation and alteration routines.

This feature requires file or system manager privileges in order to use.

Retrieve control object for a single database [NXP-60]

The `TdbDatabaseList` class has been extended with an additional constructor to make it able to return a control object for a single, named database, thesarus or cluster.

This feature requires TRIPsystem 7.0 or later for best efficiency, but will work with older versions as well.

Synchronous indexing of databases [NXP-63]

The `TdbDatabaseDesign` class has been extended with a new overloaded `Index` method that can be used to order incremental indexing and reindexing, both in batch mode and synchronously. Note that synchronous indexing may take a long time if there is a large number of records to index.

This feature requires TRIPsystem 7.0 or later in order to work.

Synchronous indexing of single records [NXP-64]

The `TdbRecord` class has been extended with a boolean property `IndexOnCommit`. By setting this to true before committing a new or updated record, the record will be synchronously indexed as part of the commit operation.

Note that batch oriented indexing remains best practise with TRIP. If synchronous indexing of single records is utilized to a great extent with a database, the scheduling of a reindex of the database from time to time is highly recommended.

This feature requires TRIPsystem 7.0 or later in order to work.

Corrected in v3.0-0

Database language information missing from `TdbDatabaseDesign` [NXP-58]

The `TdbDatabaseDesign` class does not provide any information about the natural language of the text in the database, nor is it possible to define this.

Chinese segmentation type "Word" not exposed [NXP-59]

The chinese segmentation types `All` and `Max` are supported by the `TdbDatabaseDesign` class, but the type `Word` is reported as `None`. Neither is it possible to define segmentation type `Word` for a chinese database.

Version 2.1-5

Corrected in v2.1-5

String field values may get deleted on update [NXP-61]

When updating a record containing a string field, the string field value may get deleted if it has not been altered by the application. This problem applies to TRIPnpx version 2.1-3 and 2.1-4 when used with TRIPsystem version 6.2-9:7 or newer.

TdbDateform throws exception on partial TRIP dates [NXP-62]

A TRIP DATE value may be specified with only its year component, as well as with only its year and moNth component. TRIPnpx throws an exception from the TdbDateform.FormatDate(String) method when trying to parse such a date.

Version 2.1-4

New Features in v2.1-4

Improve STRING field upload performance [NXP-53]

When uploading a value to a string field or to server-side text extraction, TRIPnpx may run out of memory alternatively time out when committing the new (or new version of a) record.

The string field value upload behavior has been changed so that large values are uploaded to the server in smaller chunks, which lowers memory and CPU consumption.

This feature requires TRIPsystem 6.2-9:8 or later to benefit from. With older versions of TRIPsystem, string value upload behavior will be the same as before.

Version 2.1-3

New Features in v2.1-3

Improve STRING field download performance [NXP-51]

When retrieving records containing large string fields, the memory usage requirements are extremely high. Fetching a single record with 100MB of binary data have been known to cause client applications to run out of memory. The same problem can manifest also with smaller string field values when retrieving a set of several records where the total size of the binary data is high.

When used with TRIPsystem version 6.2-9:7 or later, TRIPnpx will now fetch string field values on demand by utilizing a streaming mechanism added to the XPI protocol in the aforementioned version of TRIPsystem. This will lower memory usage and reduce the overall retrieval and processing time for records and search results.

This feature requires TRIPsystem 6.2-9:7 or later to benefit from. With older versions of TRIPsystem, string values will be fetched inline like before.

Corrected in v2.1-3

TdbException.DebugMessage not complete [NXP-52]

The DebugMessage property of the TdbException class would in some cases not include a stack trace. Stack traces for nested exceptions would only be one level deep.

Version 2.1-2

News in v2.1-2

TdbDatabaseDesign.OwnerName property added [NXP-46]

The TdbDatabaseDesign class has been enhanced with a property OwnerName that holds the name of the user who owns the database.

This feature requires TRIPsystem version 6.2-8:4 or later.

Filename field name for HTML conversion [NXP-49]

There are now new constructors for the classes TdbRendition and TdbHighlightRendition that take the name of a phrase field. This phrase field must contain the original name of the file stored in the string field that is about to be converted to HTML.

This information is necessary for both TRIPcof and TRIPview so that they will better be able to identify the type of the file. If the original file name is not known to the HTML converter, there is a risk that the conversion will fail.

This feature requires TRIPsystem version 6.2-9 or later.

Access rights via group memberships from TdbUserAccessList [NXP-50]

A new constructor to TdbUserAccessList has been added that takes a boolean parameter that indicates if access rights granted via group memberships also should be returned.

This feature requires TRIPsystem version 6.2-9 or later.

Corrected in v2.1-2

Invalid exception from TdbSearchSet.ClearCache() [NXP-47]

The ClearCache() method in the TdbSearchSet class would incorrectly throw an exception if called before at least one record was returned to the application via the current TdbSearchSet instance.

Server-side crash on Linux may hang TRIPnpx [NXP-48]

Sometimes when TRIP on a Linux server crashes, the system (not TRIP) will print out a message to stdout starting with "**** glibc detected ***". This is a problem with client/server sessions because the stdout stream is redirected to a communications socket. The consequence of this is that TRIPnpx receives an invalid response that is interpreted to be 539634214 bytes long. This is not correct and this mismatch caused TRIPnpx to hang indefinitely.

Version 2.1-1

News in v2.1-1

Support for search comforters [NXP-41]

Support for search comforters has been added to TRIPnpx. This makes it possible to get regular notifications ("searching...") from the server for lengthy queries, and also enables applications to abort such queries.

The example application "Search" has been extended to include comforter notification support.

This feature requires TRIPsystem 6.2-8 or later.

Corrected in v2.1-1

No output to activity log if server exits prematurely [NXP-40]

The TRIPnpx activity log would not produce any output for an XPI transaction if the server hangs or crashes. Neither request nor response would get logged.

Version 2.1-0

News in v2.1-0

Open up for segmentation type "Word" [NXP-36]

Enabled assignment of the Chinese text segmentation type "Word" to a database design. Existing database designs with this segmentation type will be correctly shown as "Word" instead of "None", which also is used in TRIPnpx for unknown or invalid segmentation values.

XPath as Query Language [NXP-37]

Enhancements have been made to the TdbSearch, TdbRecordSet, and TdbCclCommand classes so that they can be used to query TRIPxml databases using XPath expressions.

This feature requires TRIPsystem 6.2-7 or later and TRIPxml 3.0 or later. Please refer to the TRIPxml manual for valid XPath syntax.

XML Fragment Retrieval [NXP-38]

It is now possible to retrieve XML fragments from a search set containing XML records. The node set that form the roots of the fragments is specified by an XPath expression via the new TdbSearchSet method AsFragmentSet().

This feature requires TRIPsystem 6.2-7 or later and TRIPxml 3.0 or later. Please refer to the TRIPxml manual for valid XPath syntax.

Version 2.0-4

Corrected in v2.0-4

Changing password for another user does not work [NXP-34]

Being logged on as SYSTEM and using the ResetPassword method of the TdbUser class to change the password for another user did not work. Success would be reported, but the password had not changed into the specified new password. The old password would not work either after the ResetPassword call.

Changing password for current user does not work [NXP-35]

Using the ChangePassword method of the TdbUser class to change change the password of the current user did not work. The request would fail with the message "invalid arguments passed to function".

Version 2.0-3

News in v2.0-3

Control object method for checking if a database is XML enabled [R1459]

The new property IsXmlEnabled on the TdbControlObject class returns true if the database that the control object represents is XML enabled.

Enable custom ASE calls for insert and update of records [R1458]

It is now possible to assign names of ASE routines to call when updating or inserting a record. The TdbRecord class have two new properties for this: InsertAseList and UpdateAseList. Add the names of the ASE routines to call to these lists, and they get called in order.

There is a new section at the end of chapter 7 in the TRIPnpx & TRIPjxp Programmer's Guide that explains this feature.

This behavior requires TRIPsystem 6.2-4 or later.

Corrected in v2.0-3

Whitespace ignored between consecutive hit terms [B1433]

Using TdbRecordSet to issue a search order with more than one term with only whitespace between them would result in the whitespace being ignored when retrieving the value via the TdbField classes.

For example, searching for "Mad Hatter" in the ALICE demo database would return "MadHatter" in the PERSON field for the subfield in which the hit was located.

Version 2.0-2

News in v2.0-2

Automatically close idle sessions in connection pool [R1415]

It is now possible to have a connection pool automatically close sessions that have been idle for more than a specified number of seconds.

New properties have been added to the `TdbConnectionPool` class to accomplish this; `purgeInterval`, `idleTimeout` and `minIdle`.

Version 2.0-1

News in v2.0-1

Finer control when specifying what to fetch from a TEXT field [R1412]

The `TdbFieldTemplate` class has changed so that the 'Focus' property now is split into three different modes. 'Focus' means to only fetch a keyword-in-context representation of a field with hits. If no hits are available, nothing will be returned. 'Extract' means to extract a specified number of characters from the start of the field. 'FocusOrExtract' first attempts to do a keyword-in-context, but if the field contains no hits, then the specified first number of characters from the field will be returned instead. FocusOrExtract corresponds to the behavior in version 2.0-0 and earlier.

All the new fetch modes for a field is available via the new enum `TdbFieldFetchType`. The `TdbFieldTemplate` properties `Focus` and `Summary` are marked as obsolete and will be removed in the next major release.

Proper use of the `TdbFieldTemplate` class requires `TRIPsystem` version 6.2-2 or later. Earlier versions of `TRIPsystem` will also work, but the fetch types 'Extract' and 'Focus' will not work.

Version 2.0-0

News in v2.0-0

Method `TdbDatabaseClusterDesign.Remove()` added [R1401]

A `Remove()` method has been added to the `TdbDatabaseClusterDesign` class in order to make removal of cluster members easier.

Get field value size [R1373]

The total size of the value in a field can now be accessed via the `TdbField` property `FieldSize`.

If the application only needs the size of the field and not the value itself, note that a significant overhead can be incurred. In order to avoid that, there is a new property `FieldContent` on the `TdbFieldTemplate` class that if set to false, disables retrieval of the field value while retaining the capability to return the size of the field value to the application.

This behavior requires TRIPsystem 6.2-1 or later.

Access field template collection via `TdbRecord` [R1343]

The class `TdbRecord` contains a new property `FieldTemplates`, which returns an `IEnumerable<TdbFieldTemplate>` object that can be used to inspect the field templates in a retrieval template.

A usage example can be found in the new sample application "Search".

New class: `TdbSearch` [R1342]

The new class `TdbSearch` provides a search interface that combines the best of the classes in the `Ccl` and `Data` namespaces with features gathered from best practices and observations regarding ease of use of search capabilities in earlier versions of TRIPnpx.

The `TdbSearch` class is demonstrated in the new sample applications "Search" and "SearchSetIteration".

The `TdbSearch` class is EXPERIMENTAL in this release.

Microsoft .NET Framework 3.5 required [R1341]

TRIPnpx now requires version 3.5 of the .NET Framework.

FieldCount property for `TdbDatabaseDesign` [R1340]

The new `TdbDatabaseDesign.FieldCount` property provides a count of the number of fields in the database design.

Improved support for RTF output from kernel windows [R1339]

A new overloaded version of `TdbKernelWindow.ToRtfString` is now available. Together with the new methods `RtfHeader` and `RtfFooter`, it makes it possible for an application to render concatenated RTF output over multiple scroll (e.g. `ScrollDown`) operations.

The new sample "Search" makes use of this new behavior.

Generics for exposed collections [R1334]

TRIPnpx is from version 2.0 using generics for all exposed collections. A large number of properties and methods have been affected by this change. Applications are likely to require a rebuild in order to work with version 2.0 of TRIPnpx. In some cases, application code may have to be changed somewhat.

Below follows a list of changed properties and methods.

- `TdbCclCommand.HistoryUpdates`
- `TdbKernelWindow.Lines`
- `TdbKernelWindow.Hits`

- TdbTermList now inherits from List<TdbTerm> instead of ArrayList.
- TdbTermTree.Roots
- TdbTreeEntry.Parents
- TdbTreeEntry.ParentTerms
- TdbTreeEntry.Children
- TdbTreeEntry.ChildTerms
- TdbTreeEntry.Relations
- TdbTreeEntry.RelatedTerms
- TdbTreeEntry.Synonyms
- TdbCategory now implements IEnumerable<TdbTrainingInfo>
- TdbClassificationScheme now implements IEnumerable<TdbCategory>
- TdbClassificationScheme.Classify(string,IList<TdbCategory>) - type change for second parameter
- TdbControlObjectList now implements IEnumerable<TdbControlObject>
- TdbDatabaseFieldList(TdbSession,string,ICollection<int>,bool) - type change for third parameter
- TdbComponent.Fields
- TdbComponent.FieldNames
- TdbField.OriginalValues
- TdbField.Values
- TdbRecord.Parts
- TdbRecordSet.Records
- TdbRecordSet.Insert(ICollection<TdbRecord>) - type change for first parameter
- TdbStructureField.OriginalValues
- TdbStructureField.Values
- TdbTextField.OriginalValues
- TdbTextField.Values
- TdbDatabaseAccess.ReadFields
- TdbDatabaseAccess.WriteFields
- TdbDatabaseDesign.Fields
- TdbDatabaseDesign.IntegrityRules
- TdbErrorCheckPhase now inherits from List<string> instead of ArrayList.
- TdbErrorCheckSink now inherits from List<TdbErrorCheckPhase> instead of ArrayList.
- TdbFormat.Content
- TdbGridCluster.Members
- TdbGridHost.Publications
- TdbGridReplicaSet.Members
- TdbGridSysProps.Properties
- TdbBackgroundTaskSink.Messages
- TdbBackgroundTaskSink.Files
- TdbSession.Dateforms
- TdbSession.FieldTypes
- TdbSession.SubsystemNames
- TdbSession.LicensedProducts
- TdbControlObject now implements IEnumerable<string>
- TdbMessageProvider.MessageList
- TdbMessageProvider.CodeList

- TdbProcedure.Content

Support for thesaurus-based searching in TdbRecordSet [R1333]

Via the new property Thesaurus on the TdbRecordSet object, it is now possible to define a thesaurus to use with the CCL order. This makes it possible to use thesaurus-specific syntax, e.g. DOWN() and UP().

Access record timestamp through TdbRecord [R1317]

Via the new TimeStamp property on the TdbRecord class, it is now possible to get a .NET DateTime version of the timestamp for stored records.

Record ID available for new records [R1305]

When creating a new record using the TdbRecord or TdbXmlRecord classes, the RecordId property contains the record ID of the new record after successful completion of the commit() call.

TdbDateform capable of formatting TRIP time stamps [R1301]

It is now possible to use the TdbDateform class to format TRIP time stamp values such as returned by the LastUpdate property of the TdbDatabaseDesign class.

Session isolation for connection pools [R1290]

It is now possible to control the session isolation behavior for connection pools. By default, sessions in pools are isolated from each other, but this may now be disabled, so that the automatic SAVE/LOAD STATUS calls that otherwise cause network I/O are not performed.

Support for TRIPxml 2.x added [R1124]

Via the new class TdbXmlRecord, it is now possible to import and export documents to/from a TRIPxml database. This feature requires TRIPsystem version 6.2-1 or later.

Corrected in v2.0-0

Base64 routines produce invalid encoding [B1295]

The Base64 encoding is used to transmit STRING field values and output format results via the XPI protocol.

Unfortunately, the data produced by the the encoder and decoder was not valid according to the RFC 4648. In order to fix this problem, the server-side base64 routines has been modified. The client-side decoder in TRIPnpx has been consequently also been modified to reflect the necessary changes.

One of the noticeable symptoms of this were that output data could sometimes be a couple of bytes larger than what was originally sent in.

A side effect of this bug fix is that when connecting to a TRIPsystem installation older than version 6.2, the login attempt will at first fail even if a correct password was provided. TRIPnpx will then automatically fall back to the old version of the Base64 encoding and retry the login. The user will not notice anything, but the two separate login attempts can be detected on the server if logging is enabled.

Exception thrown when saving database cluster design changes [B1400]

Modifying a cluster via an instance of the `TdbDatabaseClusterDesign` class initialized via a call to `Get(string)` would cause an exception to be thrown when calling the `Put()` method to commit the changes.

Version 1.3-4

News in v1.3-4

Access to the did-you-mean facility [R1258]

In preparation for the release of TRIPsystem 6.2, the classes `TdbCclCommand` and `TdbRecordSet` have been enhanced with the property `DidYouMean`. When used with TRIPsystem 6.2 and later, any FUZZ command issued can result in a suggestion for an alternative query that is likely to give a greater amount of hits.

Version 1.3-3

Corrected in v1.3-3

Exception sometimes thrown in `TdbDatabaseAccess.Get()` for empty scopes [B1242]

The server may sometimes send an XPI response with an empty node for a read or write scope where TRIPnpx assumed the existence of a text node for the scope definition. This special case was incorrectly handled in TRIPnpx when parsing the response.

Version 1.3-2

News in v1.3-2

TRIPgrid added to `TdbProduct` enum [R1199]

TRIPgrid has been added to the `TdbProduct` enumeration, which makes it possible to check if the license on the currently connected server includes TRIPgrid.

Corrected in v1.3-2

Record name not parsed for databases not owned [B1213]

The structure of the `BASE_RES` message from the server when calling the `TdbDatabaseDesign` methods `get()` or `get(name)` is different depending on whether the logged on user is the owner of the database or not.

TRIPnpx did not pick up the record name correctly if the logged on user is not the owner of the requested database design.

Version 1.3-1

Corrected in v1.3-1

Removal of all values from a field not working [B1183]

If all values in a structured field were removed (e.g. via the `clear()` method of the `TdbField` class), no information about the removal was sent to the server upon commit. This resulted in that the field was totally unaffected, and the values remained in the field. This applies to structured fields only.

Version 1.3-0

News in v1.3-0

Tuple lists [R1151]

Tuple lists are now supported via the two new classes `TdbTupleList` and `TdbTuple` for all fields that are capable of having subfields, i.e. fields of type `Phrase`, `Integer`, `Number`, `Date` and `Time`.

There is a new section 8 in the Programmer's Guide that describes how to use tuple lists. There is also a new sample named "Tuples" that shows tuple list usage in the context of a small Windows Forms application.

Value count property for TdbField [R1153]

The property `TdbField.ValueCount` has been added that reports the number of values (subfields) in the current field.

Previously, the only way to get a subfield count was to access the `Values` property. This was not optimal, since that collection is re-created every time it is accessed.

Value insert method for TdbField [R1154]

The method `TdbField.InsertValue` has been added that enables the insertion of a value in the middle of the value list (subfields).

Value remove method for TdbField [R1155]

The method `TdbField.RemoveValue` has been added that enables the removal of a single value from a field. The only way to accomplish this prior to release 1.3-0 was to clear all values from the field, then adding the required ones back.

Date format conversion class [R1169]

There is a new class called `TdbDateform` that offers methods for conversion of `DateTime` objects to and from the current TRIP dateform. It is accessible via the `CurrentDateform` property on the `TdbSession` class.

Corrected in v1.3-0

Setting a numeric field to an empty string caused exception [B1156]

Passing an empty string to a numeric field caused TRIPnpx to throw an exception upon format validation.

Exception thrown if decimal separator in locale is not a dot [B1157]

The TRIPxpi server-side library does not accept any other decimal separator than a dot. A locale that specifies something else, e.g. a comma, caused a formatting error.

Field values passed to and obtained from the TdbNumberField class now uses the decimal separator as defined in the current locale. Any required conversion is done internally by TRIPnpx.

Version 1.2-2

News in v1.2-2

Unicode session awareness [R1145]

As per TRIPsystem 6.0-2, the TRIPxpi always starts sessions in unicode mode. The properties TdbSession.CharacterSet and TdbSession.CharacterSetName are now able to return the unicode character set enumerated type and name instead of incorrectly reporting ISOLatin1/LA1 as the fallback for an unknown character set.

Corrected in v1.2-2

Resource leaks in TdbStringField [B1039]

Exceptions thrown in the CopyFromFile and CopyToFile methods in the TdbStringField class would cause the stream to not be closed prior to return.

Exception thrown when using TdbRecordSet without a retrieval template [B1144]

Using the TdbRecordSet without a retrieval template threw an exception instead of reporting the number of records found.

Version 1.2-1

News in v1.2-1

Connection pool shutdown

A method Close() has been added to the TdbConnectionPool class. Call it to cleanly shut down the pool on application exit.

Re-establishing broken pooled sessions

The TdbConnectionPool class will now attempt to re-establish broken sessions (e.g. ones that have lost connection to the server).

Two new properties, `RestoreRetries` and `RestoreDelay`, have been added to support this behavior. The `RestoreRetries` property represents the max number of retries to use for re-establishing a broken session. The `RestoreDelay` property represents the number of milliseconds to wait between session attempts to re-establish a broken session.

Version 1.2-0

News in v1.2-0

Language support for connection pools

A connection pool can now be associated with a CCL dialect.

Connection pool limit and size

Two new properties on the `TdbConnectionPool` class (`Limit` and `Size`) allows for control of the number of sessions that can be maintained by the connection pool.

Uniform casing for method and property names

Methods and properties used a mix of pascal and camel casing. This has now been modified so that pascal casing is used exclusively.

TRIPview 2.0 HTML highlighting

The class `TdbHighlightRendition` was added to support the retrieval of highlighted HTML conversions of `STRING` field contents. Use of this feature requires TRIPview 2.0 on the server.

Property for assigned classification categories

Via the `Categories` property, the `TdbRecord` class can return an array of assigned classification category names for the record. This list will be empty unless the database is configured with a classification scheme.

Reverse retrieval of record sets

Setting the property `RetrieveReversed` on a `TdbRecordSet` instance can be used to retrieve the requested set of records in reverse order. This is equivalent of the CCL command "SHOW REVERSE".

Corrected in v1.2-0

Exception thrown when using unnamed part records

Manipulating databases with part record names using the classes in the `TietoEnator.Trip.Nxp.Data` namespace resulted in a `TdbException` being thrown with the `TdbException.INVALID_ARGS` code.

TRIPnxp assembly not marked as CLS-compliant

The TRIPnxp assembly was not marked as CLS-compliant. This resulted in CLS-compliance warnings from the compiler when building TRIPnxp-based applications.

Version 1.1-0

News in v1.1-0

New namespace name

Renamed the TRIPnpx namespaces to follow Microsoft naming guidelines. TRIPnpx namespace names all now begin with TietoEnator.Trip.Nxp.

Corrected in v1.1-0

Connection pools not working

Trying to create a connection pool resulted in a TdbException with the TdbException.UNUSABLE_SESSION code.

Known Remaining Issues

No issues are currently known.